SOLUTIONS, MATLAB Problems, Problem Set 1

18.06 Fall '12

This problem set is due Thursday, September 13, 2012 by 4pm in 2-255. The problems are out of the 4th edition of the textbook. For computational problems, please include a printout of the code with the problem set (for MATLAB in particular, `diary("filename")` will start a transcript session, `diary off` will end one, also copy and paste usually work as well.)

7. **Q:** (This computational problem will ask you to open up your favorite computational package, and figure out how to enter a matrix, to matrix multiply, and take the transpose of a matrix. It also asks you to find a pattern.) The 4x4 Pascal Matrix is

$$P = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 1 & 3 & 6 & 10 \\ 1 & 4 & 10 & 20 \end{pmatrix}.$$

Look up Pascal's triangle, if you have never heard of it before. A closely related triangle is the lower triangular 4x4 Pascal Matrix

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 2 & 1 & 0 \\ 1 & 3 & 3 & 1 \end{pmatrix}.$$

Verify on the computer that $P = LL^T$. (L times the transpose of L.) Continue the pattern and create the 5x5 P and the 5x5 L on the computer and verify again that $P = LL^T$ in the 5x5 case.

7. **A:** The 4-by4 Pascal matrix can be produced either with the command

```
>> P = pascal(4);
```

or

```
>> P = [ 1,1,1,1; 1,2,3,4; 1,3,6,10; 1,4,10,20 ]
P =

     1     1     1     1
     1     2     3     4
     1     3     6    10
     1     4    10    20
```

Next the lower triangular factor:

```
>> L = [ 1,0,0,0; 1,1,0,0; 1,2,1,0; 1,3,3,1]

L =
```

```
     1     0     0     0
     1     1     0     0
     1     2     1     0
     1     3     3     1
```

(One could also use MATLAB's Cholesky factorization: `>> L = chol( P )'`)

Then check `L` times its transpose:

```
>> L*L'

ans =

     1     1     1     1
     1     2     3     4
     1     3     6    10
     1     4    10    20
```

(Recall that MATLAB uses ' for transpose.)

Repeating for the 5-by-5 case:

```
>> P = pascal(5)

P =
     1     1     1     1     1
     1     2     3     4     5
     1     3     6    10    15
     1     4    10    20    35
     1     5    15    35    70

>> L = [ 1,0,0,0,0; 1,1,0,0,0; 1,2,1,0,0; 1,3,3,1,0; 1,4,6,4,1 ]

L =

     1     0     0     0     0
     1     1     0     0     0
     1     2     1     0     0
     1     3     3     1     0
     1     4     6     4     1

>> L*L'

ans =
     1     1     1     1     1
     1     2     3     4     5
     1     3     6    10    15
     1     4    10    20    35
     1     5    15    35    70
```

9. **Q:** (This problem is an investigation on a computer.) Create an identity matrix (for example, in MATLAB, `I=eye(5)`) and a permutation vector (example `p=[3 4 1 2 5]`). Create a permutation matrix by permuting the columns (example `P=I(:,p)`). Compute matrix powers (`P,P^2,P^3,P^4,...`) Which is the smallest positive power that returns P to the identity? Find five different p's, each with the property that $P^k = I$, for $k = 1, 2, 3, 4, 5$ and k is the smallest such value. Hint: when $k = 1$, the answer is p=[1 2 3 4 5]. When k=5, the answer is p=[2 3 4 5 1]. Now you should find the answers when k=2,3, and 4.

9. **A:** Answers to this one will vary. Here is one approach.
First create an identity matrix and a permutation vector:

```
>> I = eye(5)

I =

     1     0     0     0     0
     0     1     0     0     0
     0     0     1     0     0
     0     0     0     1     0
     0     0     0     0     1

>> p = [1,2,4,5,3]

p =
     1     2     4     5     3
```

(I just picked `p` at random, others will do.)
Next, use these to make a permutation matrix:

```
>> P = I(:,p)

P =

     1     0     0     0     0
     0     1     0     0     0
     0     0     0     0     1
     0     0     1     0     0
     0     0     0     1     0
```

Now successively raise to `P` to larger powers until the result is `I`:

```
>> P^2

ans =

     1     0     0     0     0
     0     1     0     0     0
     0     0     0     1     0
     0     0     0     0     1
```

```
     0     0     1     0     0

>> P^3

ans =

     1     0     0     0     0
     0     1     0     0     0
     0     0     1     0     0
     0     0     0     1     0
     0     0     0     0     1
```

To find P's that return $\texttt{P}^k$ =I for $k = 2, 3, 4$, there are a few approaches. One is trial and error, which I'll leave to you.

A second is to use some reasoning. For example, in the case above, P leaves the first two rows alone and cycles rows 3 through 5. In other words, successive application of P goes like [1,2,4,5,3] → [1,2,5,3,4] → [1,2,3,4,5].

So to get $k = 2$, we could try a P that fixes 3 rows and cycles 2 rows:

```
>> p = [1,2,3,5,4]

p =
     1     2     3     5     4

>> P = I(:,p)

P =
     1     0     0     0     0
     0     1     0     0     0
     0     0     1     0     0
     0     0     0     0     1
     0     0     0     1     0

>> P^2

ans =

     1     0     0     0     0
     0     1     0     0     0
     0     0     1     0     0
     0     0     0     1     0
     0     0     0     0     1
```

Simlarly, to get $k = 4$, we could try a P that fixes 1 row and cycles 4 rows:

```
>> p = [1,5,2,3,4]

p =
```

```
     1     5     2     3     4

>> P = I(:,p)

P =
     1     0     0     0     0
     0     0     1     0     0
     0     0     0     1     0
     0     0     0     0     1
     0     1     0     0     0

>> P^2

ans =

     1     0     0     0     0
     0     0     0     1     0
     0     0     0     0     1
     0     1     0     0     0
     0     0     1     0     0

>> P^3

ans =

     1     0     0     0     0
     0     0     0     0     1
     0     1     0     0     0
     0     0     1     0     0
     0     0     0     1     0
>> P^4

ans =

     1     0     0     0     0
     0     1     0     0     0
     0     0     1     0     0
     0     0     0     1     0
     0     0     0     0     1
```